




# Tight Bounds for Counting Colorings and Connected Edge Sets Parameterized by Cutwidth

Carla Groenland   



Utrecht University, The Netherlands

Isja Mannens  

Utrecht University, The Netherlands

Jesper Nederlof  

Utrecht University, The Netherlands

Krisztina Szilágyi  

Utrecht University, The Netherlands

---

## Abstract

We study the fine-grained complexity of counting the number of colorings and connected spanning edge sets parameterized by the cutwidth and treewidth of the graph. While decompositions of small treewidth decompose the graph with small vertex separators, decompositions with small cutwidth decompose the graph with small *edge* separators.

Let  $p, q \in \mathbb{N}$  such that  $p$  is a prime and  $q \geq 3$ . We show:

- If  $p$  divides  $q - 1$ , there is a  $(q - 1)^{\text{ctw}} n^{O(1)}$  time algorithm for counting list  $q$ -colorings modulo  $p$  of  $n$ -vertex graphs of cutwidth  $\text{ctw}$ . Furthermore, there is no  $\varepsilon > 0$  for which there is a  $(q - 1 - \varepsilon)^{\text{ctw}} n^{O(1)}$  time algorithm that counts the number of list  $q$ -colorings modulo  $p$  of  $n$ -vertex graphs of cutwidth  $\text{ctw}$ , assuming the Strong Exponential Time Hypothesis (SETH).
- If  $p$  does not divide  $q - 1$ , there is no  $\varepsilon > 0$  for which there exists a  $(q - \varepsilon)^{\text{ctw}} n^{O(1)}$  time algorithm that counts the number of list  $q$ -colorings modulo  $p$  of  $n$ -vertex graphs of cutwidth  $\text{ctw}$ , assuming SETH.

The lower bounds are in stark contrast with the existing  $2^{\text{ctw}} n^{O(1)}$  time algorithm to compute the chromatic number of a graph by Jansen and Nederlof [Theor. Comput. Sci.'18].

Furthermore, by building upon the above lower bounds, we obtain the following lower bound for counting connected spanning edge sets: there is no  $\varepsilon > 0$  for which there is an algorithm that, given a graph  $G$  and a cutwidth ordering of cutwidth  $\text{ctw}$ , counts the number of spanning connected edge sets of  $G$  modulo  $p$  in time  $(p - \varepsilon)^{\text{ctw}} n^{O(1)}$ , assuming SETH. We also give an algorithm with matching running time for this problem.

Before our work, even for the treewidth parameterization, the best conditional lower bound by Dell et al. [ACM Trans. Algorithms'14] only excluded  $2^{o(\text{tw})} n^{O(1)}$  time algorithms for this problem.

Both our algorithms and lower bounds employ use of the matrix rank method, by relating the complexity of the problem to the rank of a certain “compatibility matrix” in a non-trivial way.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Parameterized complexity and exact algorithms

**Keywords and phrases** connected edge sets, cutwidth, parameterized algorithms, colorings, counting modulo  $p$

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2022.36

**Related Version** *Full Version*: <https://arxiv.org/abs/2110.02730>

**Funding** All authors are supported by the project CRACKNP that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 853234).



**Acknowledgements** We thank the anonymous reviews for their detailed comments.



© Carla Groenland, Isja Mannens, Jesper Nederlof, and Krisztina Szilágyi; licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022).

Editors: Petra Berenbrink and Benjamin Monmege; Article No. 36; pp. 36:1–36:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

A popular topic of interest in (fine-grained) algorithmic research is to determine the decomposability of NP-hard problems in easier subproblems. A natural decomposition strategy is often implied by decomposing the solution into sub-solutions induced by a given decomposition of the input graph such as tree decompositions, path decompositions, or tree depth decompositions, independent of the problem to be solved. However, the efficiency of such a decomposition can wildly vary per computational problem. Recently, researchers developed tools that allow them to get a precise understanding of this efficiency: non-trivial algorithmic tools (such as convolutions and the cut-and-count method [9, 25]) were developed to give algorithms that have an optimal running time conditioned on hypotheses such as the Strong Exponential Time Hypothesis (SETH) [16]. While the efficiency of such decompositions has been settled for most decision problems parameterized by treewidth, many other interesting settings remain elusive. Two of them are *cutwidth* and *counting problems*.

The *cutwidth* of an ordering of the vertices of the graph is defined as the maximum number of edges with exactly one endpoint in a prefix of the ordering (where the maximum is taken over all prefixes of the ordering). The *cutwidth* of a graph is defined to be the minimum width over all its cutwidth orderings. Cutwidth is very similar to pathwidth, except that cutwidth measures the number of edges of a cut, while the pathwidth measures the number of *endpoints* of edges over the cut. Thus the cutwidth of a graph is always larger than its pathwidth. But for some problems a decomposition scheme associated with a cutwidth ordering of cutwidth  $k$  can be used much more efficiently than a decomposition of pathwidth  $k$ . A recent example of such a problem is the  $q$ -coloring problem:<sup>1</sup> While there is a  $(q - \varepsilon)^{pw}$  lower bound [21] assuming SETH, there is a  $2^{ctw} n^{O(1)}$  time randomized algorithm [18].

*Counting problems* pose an interesting challenge if we want to study their decomposability. Counting problems are naturally motivated if we are interested in any statistic rather than just existence of the solutions space. While often a counting problem behaves very similarly to its decision version (as in, the dynamic programming approach can be fairly directly extended to solve the counting version as well), for some problems there is a rather puzzling increase in complexity when going from the decision version to the counting version.<sup>2</sup>

One of the most central problems in counting complexity is the evaluation of the *Tutte polynomial*. The strength of this polynomial is that it expresses all graph invariants that can be written as a linear recurrence using only the edge deletion and contraction operation [23], and its evaluations specialize to a diverse set of parameters ranging from the number of forests, nowhere-0 flows,  $q$ -colorings and spanning connected edge sets.

An interesting subdirection within counting complexity that is in between the decision and counting version and that we will also address in this paper is *modular counting*, where we want to count the number of solutions modulo a number  $p$ . This is an interesting direction since the complexity of the problem at hand can wildly vary for different  $p$  (see [24] for a famous example), but in the setting of this paper it is also naturally motivated: For example, the cut-and-count method achieves the fastest algorithms for several decision problems by actually solving the modular counting variant instead.

<sup>1</sup> Recall that a  $q$ -coloring is a mapping from the vertices of the graph to  $\{1, \dots, q\}$  such that every two adjacent vertices receive distinct colors, and the  $q$ -coloring problem asks whether a  $q$ -coloring exists.

<sup>2</sup> Two examples herein are detecting/counting perfect matchings (while the decision version is in  $P$ , the counting version can not be solved in time  $(2 - \varepsilon)^{tw} n^{O(1)}$  for any  $\varepsilon > 0$  assuming the SETH [7]) and Hamiltonian cycles (while the decision version can be solved in  $(2 + \sqrt{2})^{pw}$  time [8], the counting version can not be solved in time  $(6 - \varepsilon)^{tw} n^{O(1)}$  for any  $\varepsilon > 0$  assuming the SETH [6]).

## 1.1 Our results

In this paper we study the complexity of two natural hard (modular) counting problems: Counting the number of  $q$ -colorings of a graph and counting the number of spanning connected edge sets, parameterized by the cutwidth of the graph.

**Counting Colorings.** Let  $G$  be a graph and suppose that for each  $v \in V$  we have an associated list  $L(v) \subseteq \{1, \dots, q\}$ . A *list  $q$ -coloring* is a coloring  $c$  of  $G$  such that  $c(v) \in L(v)$  for each  $v \in V$ . Two colorings are *essentially distinct* if they cannot be obtained from each other by permuting the color classes. Since the number of essentially distinct colorings is  $q!$  times the number of distinct colorings (assuming the chromatic number of the graph is  $q$ ), counting colorings modulo  $p$  may become trivial if  $p \leq q$ . For this reason, we focus on counting essentially distinct colorings in our lower bounds.

In this paper, we will focus on counting list  $q$ -colorings modulo a prime number  $p$ . Our main theorem reads as follows:

► **Theorem 1.** *Let  $p, q \in \mathbb{N}$  with  $p$  prime and  $q \geq 3$ .*

- *If  $p$  divides  $q-1$ , then there is a  $(q-1)^{\text{ctw}} n^{O(1)}$  time algorithm for counting list  $q$ -colorings modulo  $p$  of  $n$ -vertex graphs of cutwidth  $\text{ctw}$ . Furthermore, there is no  $\varepsilon > 0$  for which there exists a  $(q-1-\varepsilon)^{\text{ctw}} n^{O(1)}$  time algorithm that counts the number of essentially distinct  $q$ -colorings modulo  $p$  in time  $(q-1-\varepsilon)^{\text{ctw}} n^{O(1)}$ , assuming SETH.*
- *If  $p$  does not divide  $q-1$ , there is no  $\varepsilon > 0$  for which there exists a  $(q-\varepsilon)^{\text{ctw}} n^{O(1)}$  time algorithm that counts the number of essentially distinct  $q$ -colorings modulo  $p$ , assuming SETH.*

Thus, we show that under the cutwidth parameterization, the (modular) counting variant of  $q$ -coloring is much harder than the decision, as the latter can be solved in  $2^{\text{ctw}} n^{O(1)}$  time with a randomized algorithm [18]. Additionally, we show there is a curious jump in complexity based on whether  $p$  divides  $q-1$  or not: Since our bounds are tight, this jump is inherent to the problem and not an artifact of our proof.

The proof strategy of all items of Theorem 1 relates the complexity of the problems to a certain *compatibility matrix*. This is a Boolean matrix that has its rows and columns indexed by partial solutions, and has a 1 if and only if the corresponding partial solutions combine into a global solution. In previous work, it was shown that the rank of this matrix can be used to design both algorithms [4, 8, 18, 22] and lower bounds [6, 8].

With this in mind, the curious jump can intuitively be explained as follows. Consider the base case where the graph is a single edge and we decompose a (list)  $q$ -coloring into the two colorings induced on the vertices. The compatibility matrix corresponding to this decomposition is the complement of an  $q \times q$  identity matrix. This matrix has full rank if  $p$  does not divide  $q-1$  and it has rank  $q-1$  otherwise. We believe this is a very clean illustration of the rank based methods, since it explains a curious gap that would be rather mysterious without the rank based perspective.

**Connected Spanning Edge Sets and Tutte polynomial.** We say that  $X \subseteq E$  is a *connected spanning edge set* if  $G[X]$  is connected and every vertex is adjacent to an edge in  $X$ . Our second result is about counting the number of such sets. This problem is naturally motivated: It gives the probability that a random subgraph remains connected, and is an important special case of the Tutte polynomial. We determine the complexity of counting connected spanning edge sets by treewidth and cutwidth by giving matching lower and upper bounds:

► **Theorem 2.** *Let  $p$  be a prime number. There is an algorithm that counts the number of connected edge sets modulo  $p$  of  $n$ -vertex graphs of treewidth  $tw$  in time  $p^{tw}n^{O(1)}$ .*

*Furthermore, there is no  $\varepsilon > 0$  for which there is an algorithm that counts the number of spanning connected edge sets modulo  $p$  of  $n$ -vertex graphs of cutwidth  $ctw$  in time  $(p - \varepsilon)^{ctw}n^{O(1)}$ , assuming SETH.*

Note that before our work, even for the treewidth parameterization, the best conditional lower bound by Dell et al. [10] only excluded  $2^{o(tw)}n^{O(1)}$  time algorithms for this problem.

While the algorithm follows relatively quickly by using a cut-and-count type dynamic programming approach, obtaining the lower bound is much harder.

In fact, for related counting variants of connectivity problems such as counting the number of Hamiltonian cycles or Steiner trees,  $2^{O(tw)}n^{O(1)}$  time algorithms do exist. So one may think that connected spanning edge sets can be counted in a similar time bound. But in Theorem 2 we show that this is not the case (by choosing  $p$  arbitrarily large).

To prove the lower bound, we make use of an existing formula for the Tutte polynomial that relates the number of connected spanning edge sets to the number of essentially distinct colorings, and subsequently apply Theorem 1.

**Organization.** The rest of the paper is organized as follows: in Section 2 we introduce the notation that will be used throughout the paper and define the color compatibility matrix. In Section 3 we prove the upper bound for  $\#q$ -coloring modulo  $p$ . Section 4 contains the results about lower bounds. We conclude the paper by discussing directions for further research. The appendix contains the proofs omitted from previous sections.

## 1.2 Related work

**Coloring.** Counting the number of colorings of a graph is known to be  $\#P$ -complete, even for special classes of graphs such as triangle free regular graphs [14]. Björklund and Husfeldt [2] and Koivisto [20] gave a  $2^n n^{O(1)}$  algorithm for counting  $q$ -colorings, and a more general  $2^n n^{O(1)}$  time algorithm even evaluates any point of the Tutte polynomial [3].

A  $q$ -coloring of a graph  $G$  is a special case of  $H$ -coloring, i.e. a homomorphism from  $G$  to a given graph  $H$ . Namely,  $q$ -colorings correspond to homomorphisms from  $G$  to  $K_q$ , i.e.  $K_q$ -colorings. Dyer and Greenhill [11] showed that counting the number of  $H$ -colorings is  $\#P$ -complete unless  $H$  is one of the few exceptions (an independent set, a complete graph with loops on every vertex or a complete bipartite graph). Kazemina and Bulatov [19] classified the hardness of counting  $H$ -colorings modulo a prime  $p$  for square-free graphs  $H$ .

**Methods.** Our approach makes use of the rank based method, and in particular the so-called color compatibility matrix introduced in [18]. This matrix tells us whether we can “combine” two colorings. In [18], the authors studied the rank of a different matrix with the same support as the color compatibility matrix, whereas in this paper we use the rank directly. The rank based method has been used before only once for an algorithm for a counting problem in [8] and only once for a lower bound for a counting problem in [6].

The Tutte polynomial  $T(G; x, y)$  is a graph polynomial in two variables which describes how  $G$  is connected. In particular, calculating  $T(G; x, y)$  at specific points gives us the number of subgraphs of  $G$  with certain properties:  $T(G; 2, 1)$  is equal to the number of forests in  $G$ ,  $T(G; 1, 1)$  is the number of spanning forests,  $T(G; 1, 2)$  counts the number of spanning connected subgraphs etc. We will use the properties of the Tutte polynomial to give a lower bound on the complexity of counting spanning connected edge sets.

## 2 Preliminaries

In this section, we introduce the notation that will be used throughout the paper.

### 2.1 Notation and standard definitions

For integers  $a, b$ , we write  $[a, b] = \{a, a + 1, \dots, b\}$  for the integers between  $a$  and  $b$ , and for a natural number  $n$  we short-cut  $[n] = [1, n] = \{1, \dots, n\}$ . Throughout the paper,  $p$  will denote a prime number and  $\mathbb{F}_p$  the finite field of order  $p$ . We will use  $a \equiv_p b$  to denote that  $a$  and  $b$  are congruent modulo  $p$ , i.e. that  $p$  divides  $a - b$ . We write  $\mathbb{N} = \{1, 2, \dots\}$  for the set of natural numbers.

For a function  $f : A \rightarrow \mathbb{Z}$  (where  $A$  is any set), we define the support of  $f$  as the set  $\text{supp}(f) = \{a \in A : f(a) \neq 0\}$ . For  $B \subseteq A$ , the function  $f|_B : B \rightarrow \mathbb{Z}$  is defined as  $f|_B(b) = f(b)$  for all  $b \in B$ .

In this paper, all graphs will be undirected and simple. Given a graph  $G = (V, E)$  and a vertex  $v \in V$ , we denote by  $N(v)$  the open neighbourhood of  $v$ , i.e. the set of all vertices adjacent to  $v$ . We often use  $n$  for the number of vertices of  $G$ , and denote the cutwidth of  $G$  by  $\text{ctw}$ . We sometimes write  $V(G)$  for the vertex set of the graph  $G$ .

Note that, if  $G$  is not connected, we can count the number of  $q$ -colorings in each connected component and multiply them to get the total number of  $q$ -colorings of  $G$ . Therefore, we may assume that  $G$  is connected.

Given a graph  $G = (V, E)$ , and lists  $L : V \rightarrow 2^{[q]}$ , a *list  $q$ -coloring* of  $G$  is a coloring  $c : V \rightarrow [q]$  of its vertices such that  $c(u) \neq c(v)$  for all edges  $uv$  and  $c(v) \in L(v)$  for all vertices  $v$ . We will often abbreviate “list  $q$ -coloring” to “coloring”. For a subset  $B \subseteq V(G)$ , we will use the abbreviation  $c(B) = \{c(v) : v \in B\}$ .

Cutwidth and treewidth are graph parameters which are often used in parameterized complexity. Informally, treewidth describes how far a graph is from being a tree. The cutwidth is defined as follows.

► **Definition 3.** *The cutwidth of a graph  $G$  is the smallest  $k$  such that its vertices can be arranged in a sequence  $v_1, \dots, v_n$  such that for every  $i \in [n - 1]$ , there are at most  $k$  edges between  $\{v_1, \dots, v_i\}$  and  $\{v_{i+1}, \dots, v_n\}$ .*

We recall the definition of Tutte polynomial.

► **Definition 4.** *For a graph  $G$ , we denote by  $T(G; x, y)$  the Tutte polynomial of  $G$  evaluated at the point  $(x, y)$ . If  $G$  has no edges we have  $T(G; x, y) = 1$ . Otherwise we have*

$$T(G; x, y) = \sum_{A \subseteq E(G)} (x - 1)^{r(E) - r(A)} (y - 1)^{|A| - r(A)}$$

where  $r(A) = |V(G)| - k(A)$  indicates the rank of the edge set  $A$  and  $k(A)$  indicates the number of connected components of  $(V, A)$ .

Note that  $T(G; 1, 2)$  is exactly the number of spanning connected edge sets.

We denote the counting version of a problem by using the prefix #, and the counting modulo  $p$  version of by using  $\#_p$  (e.g.  $\#_p\text{SAT}$ ,  $\#_p\text{CSP}$ ).

### 2.2 The color compatibility matrix and its rank

Given a subset  $A \subseteq V$ , we use  $\text{col}_L(A)$  to denote the set of all list  $q$ -colorings of  $G[A]$ . If it is clear which lists are used, we omit the subscript.

## 36:6 Tight Bounds for Counting Colorings and Connected Edge Sets

It is often useful to color parts of the graph separately, and then “combine” those colorings. If two colorings can be combined without conflicts, we call them compatible:

► **Definition 5.** For subsets  $A, B \subseteq V$  and colorings  $x \in \text{col}(A)$ ,  $z \in \text{col}(B)$ , we say that  $x$  and  $z$  are compatible, written  $x \sim z$ , if

- $x(v) = z(v)$  for all  $v \in A \cap B$ , and
- $x(u) \neq z(v)$  for all  $uv \in E$ , where  $u \in A$  and  $v \in B$ .

For a set of colorings  $\mathcal{S} \subseteq \text{col}(B)$ , we write  $\mathcal{S}[x]$  for the set of colorings  $y \in \mathcal{S}$  that are compatible with  $x$ .

If  $x \sim z$ , then we define  $x \cup z$  as the  $q$ -list coloring of  $G[A \cup B]$  such that  $(x \cup z)(a) = x(a)$  for all  $a \in A$  and  $(x \cup z)(b) = z(b)$  for all  $b \in B$ . This is well-defined by the definition above.

A key definition for this paper is the following.

► **Definition 6.** Let  $(X \cup Y, E)$  be a bipartite graph and  $q$  a natural number. The  $q$ th color compatibility matrix  $M$  is indexed by all  $q$ -colorings of  $X$  and  $Y$ , with

$$M[x, y] = \begin{cases} 1, & \text{if } x \sim y, \\ 0, & \text{otherwise,} \end{cases}$$

for  $x \in \text{col}(X)$  and  $y \in \text{col}(Y)$ .

We denote the color compatibility matrix indexed by all  $q$ -colorings associated with the bipartite graph that is matching on  $t$  vertices by  $J_t$ , and short-hand  $J := J_1$ .

We will show that, if  $p$  divides  $q - 1$ , we can count all  $q$ -list colorings modulo  $p$  more quickly due to the following bound on the rank of the color compatibility matrices.

► **Lemma 7.** Let  $p$  be a prime,  $q$  a natural number and let  $G = (X \cup Y, E)$  be a bipartite graph with  $q$ th color compatibility matrix  $M$ . Then the rank of  $M$  over  $\mathbb{F}_p$  satisfies

$$\text{rank}_p(M) \leq \begin{cases} (q - 1)^{|E|} & \text{if } p \text{ divides } q - 1, \\ q^{|E|} & \text{otherwise.} \end{cases}$$

Moreover, equality is achieved if  $G$  is a perfect matching.

The proof can be found in the full version of our paper [15].

In particular,  $J_t$  is invertible mod  $p$  if and only if  $p$  does not divide  $q - 1$ .

### 3 Algorithm for # $q$ -coloring modulo $p$

In this section we prove the first part of the first item of Theorem 1:

► **Theorem 8.** Let  $G$  be a graph with  $n$  vertices and cutwidth  $\text{ctw}$ . Given an integer  $q \geq 3$  and a prime  $p$  that divides  $q - 1$ , there is an  $(q - 1)^{\text{ctw}} n^{O(1)}$  algorithm for counting list  $q$ -colorings modulo  $p$ .

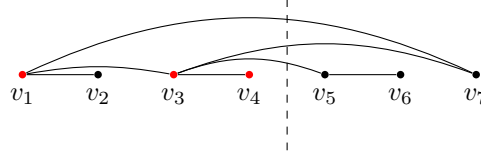
#### 3.1 Definitions and overview

We first introduce some additional notation and definitions needed in this section. Let  $q$  be an integer and let  $p$  be a prime that divides  $q - 1$ . We are given a graph  $G = (V, E)$  with the cutwidth ordering  $v_1, \dots, v_n$  of the vertices. Without loss of generality, we may assume that  $G$  is connected. We write  $G_i = G[\{v_1, \dots, v_i\}]$  and

$$L_i = \{v \in V(G_i) : vv_j \in E \text{ for some } j > i\}.$$

Note that by definition of cutwidth,  $L_i \subseteq L_{i-1} \cup \{v_i\}$  and  $|L_i| \leq \text{ctw}$  for all  $i$  (since the number of endpoints of a set of edges is upper bounded by the number of edges in the set).

Let  $i \in [n]$  be given and write  $X_i = L_i \cup \{v_i\}$  for the set of vertices left of the cut that either have an edge in the cut, or are the rightmost vertex left of the cut. We also define  $Y_i = \{v_{i+1}, \dots, v_n\} \cap N(X_i)$ . Figure 1 illustrates this notation.



■ **Figure 1** In the above graph,  $L_4 = \{v_1, v_3\}$ ,  $X_4 = \{v_1, v_3, v_4\}$  (the red vertices) and  $Y_4 = \{v_5, v_7\}$ .

Let  $T_i[x]$  be the number of extensions of  $x \in \text{col}(X_i)$  to a coloring of  $G_i = G[\{v_1, \dots, v_i\}]$ . Equivalently,  $T_i[x]$  gives the number of colorings of  $G_i$  that are compatible with  $x$ .

A standard dynamic programming approach builds on the following observation.

► **Lemma 9 (Folklore).** For  $x \in \text{col}(X_i)$ ,

$$T_i[x] = \sum_{\substack{z \in \text{col}(X_{i-1}) \\ z \sim x}} T_{i-1}[z].$$

The proof can be found in the full version of our paper [15]. Since  $|\text{col}(X_i)|$  may be of size  $q^{|X_i|}$ , we cannot compute  $T_i$  in its entirety within the claimed time bound. The idea of our algorithm is to use the same dynamic programming iteration, but to compute the values of  $T_i$  only for a subset  $\mathcal{S}'_i \subseteq \text{col}(X_i)$  of the possible colorings which is of significantly smaller size. In fact, we will compute a function  $T'_i : \mathcal{S}'_i \rightarrow \mathbb{F}_p$  that does not necessarily agree with  $T_i$  on  $\mathcal{S}'_i$ . The important property that we aim to maintain is that  $T'_i$  carries the “same information” about the number of colorings modulo  $p$  as  $T_i$  does. This is formalised below.

► **Definition 10.** Let  $H = (X \cup Y, E)$  be a bipartite graph with color compatibility matrix  $M$ . Let  $T, T' : \text{col}(X) \rightarrow \mathbb{F}_p$ . We say  $T'$  is an  $M$ -representative of  $T$  if

$$\sum_{x \in \text{col}(X)} M[x, y]T[x] \equiv_p \sum_{x \in \text{col}(X)} M[x, y]T'[x] \text{ for all } y \in \text{col}(Y).$$

In other words,  $T'$  is an  $M$ -representative of  $T$  if  $M^\top \cdot T \equiv_p M^\top \cdot T'$ .

Above we left the lists and the integer  $q$  implicit. We recall that the color compatibility matrix has entries  $M[x, y] = 1$  if  $x \in \text{col}(X)$  and  $y \in \text{col}(Y)$  are compatible, and  $M[x, y] = 0$  otherwise. Let  $i \in [n - 1]$  be given. Let  $M_i$  be the color compatibility matrix of the bipartite graph given by the edges between  $X_i$  and  $Y_i$ .

Then for  $y \in \text{col}(Y_i)$ ,

$$\sum_{x \in \text{col}(X_i)} M_i[x, y]T_i[x]$$

gives the number of colorings of  $G_i$  compatible with  $y$ . If we can compute  $T'_{n-1}$  that is an  $M_{n-1}$ -representative of  $T_{n-1}$ , then by Lemma 9 we can compute the number of  $q$ -list colorings of the graph (modulo  $p$ ) as

$$\sum_{y \in \text{col}(G[v_n])} \sum_{x \in \text{supp}(T'_{n-1})} M_{n-1}[x, y]T'_{n-1}[x].$$

It is an exercise in linear algebra to show that there always exists a  $T'$  that  $M$ -represents  $T$  with  $|\text{supp}(T')| \leq \text{rank}(M)$ . We also need to make sure that we can actually compute this  $T'$  within the desired time complexity and therefore reduce the support in a slightly more complicated fashion in Section 3.2. We then prove an analogue of Lemma 9 in Section 3.3, and describe our final algorithm in Section 3.4.

### 3.2 Computing a reduced representative

In this subsection, we show how to find a function  $T'$  that  $M$ -represents  $T$ , while decreasing an upper bound on the size of the support of the function.

► **Definition 11.** For a function  $f : \text{col}(X) \rightarrow \mathbb{F}_p$  we say that  $r \in X$  is a reduced vertex if  $f(c) = 0$  whenever  $c(r) = q$ .

The link between reduced vertices and the support of  $T : \text{col}(X) \rightarrow \mathbb{F}_p$  is explained as follows. If  $R$  is a set of reduced vertices of  $T$ , then we can compute a set of colorings containing the support of  $T$  of size at most  $(q-1)^{|R|}q^{|\text{col}(X)|-|R|}$ . Indeed, we may restrict to the colorings that do not assign the color  $q$  to any vertex in  $R$ .

The following result allows us to turn vertices of degree 1 in  $H$  into reduced vertices. The assumption that the vertex has degree 1 will be useful in proving the result because it implies the associated compatibility matrix can be written as a Kronecker product with  $J_q$  and another matrix.

► **Lemma 12.** There is an algorithm **Reduce** that, given a bipartite graph  $H$  with parts  $X, Y$  and associated color compatibility matrix  $M$ , a function  $T : \text{col}(X) \rightarrow \mathbb{F}_p$  with reduced vertices  $R \subseteq X$  and a vertex  $v \in X \setminus R$  of degree 1 in  $H$ , outputs a function  $T' : \text{col}(X) \rightarrow \mathbb{F}_p$  with reduced vertices  $R \cup \{v\}$  that is an  $M$ -representative of  $T$ . The run time is in  $O((q-1)^{|R|}q^{|\text{col}(X)|-|R|})$ .

The proof is given in Appendix A. We say that a function  $T : \text{col}(X) \rightarrow \mathbb{F}_p$  is *fully reduced* if every vertex  $v \in X$  of degree 1 is a reduced vertex of  $T$ . In order to keep the running time low, we will ensure that  $R$  is relatively large whenever we apply Lemma 12.

### 3.3 Computing $T'_i$ from $T'_{i-1}$

Recall that  $T_i[x]$  gives the number of colorings of  $G_i$  that are compatible with  $x \in \text{col}(X_i)$  and that  $M_i$  is the color compatibility matrix of the bipartite graph between  $X_i$  and  $Y_i$  (corresponding to the  $i$ th cut).

► **Lemma 13.** Let  $i \in [n-1]$ . Suppose that  $T'_{i-1}$  is an  $M_{i-1}$ -representative of  $T_{i-1}$  and that  $T'_{i-1}$  is fully reduced. Given  $T'_{i-1}$  and a set  $R_{i-1}$  of reduced vertices for  $T'_{i-1}$ , we can compute a function  $T'_i$  that is an  $M_i$ -representative of  $T_i$  in time  $O((q-1)^{|R_{i-1}|}q^{|\text{col}(X_{i-1})|-|R_{i-1}|+1})$ , along with a set  $R_i$  of reduced vertices for  $T'_i$  such that  $|X_i \setminus R_i| \leq (\text{ctw} - |R_i|)/2 + 1$ .

**Proof.** Let  $i \in [n-1]$  and let  $T'_{i-1}$  be  $M_{i-1}$ -representative of  $T_{i-1}$  and fully reduced, with  $R_{i-1}$  a set of reduced vertices for  $T'_{i-1}$ . We need to compute (in time  $O((q-1)^{|R_{i-1}|}q^{|\text{col}(X_{i-1})|-|R_{i-1}|+1})$ ) a function  $T'_i$  that is  $M_i$ -representative of  $T_i$ , along with a set  $R_i$  of reduced vertices for  $T'_i$ , such that  $|X_i \setminus R_i| \leq (\text{ctw} - |R_i|)/2 + 1$ .

We will work over  $\mathbb{F}_p$  during this proof, in particular abbreviating  $\equiv_p$  to  $=$ . Analogous to Lemma 9, we define, for  $x \in \text{col}(X_i)$ ,

$$T'_i[x] = \sum_{\substack{z \in \text{col}(X_{i-1}) \\ z \sim x}} T'_{i-1}[z]. \quad (1)$$

Note that

$$\sum_{\substack{z \in \text{col}(X_{i-1}) \\ z \sim x}} T'_{i-1}[z] = \sum_{\substack{z \in \text{supp}(T'_{i-1}) \\ z \sim x}} T'_{i-1}[z].$$



We compute  $T'_i$  from  $T'_{i-1}$  as follows. Let

$$\mathcal{S}'_{i-1} = \{c \in \text{col}(X_{i-1}) : c(r) \neq q \text{ for all } r \in R_{i-1}\}.$$

By the definition of reduced vertex,  $\mathcal{S}'_{i-1}$  contains the support of  $T'_{i-1}$  since  $T'_{i-1}$  is fully reduced. Recall that  $X_i \setminus \{v_i\} \subseteq X_{i-1}$ , so any  $x \in \text{col}(X_i)$  is determined if we provide colors for the vertices in  $X_{i-1} \cup \{v_i\}$ . For a color  $c \in [q]$ , let  $f_c : \{v_i\} \rightarrow \{c\}$  be the function that assigns color  $c$  to  $v_i$ . For each  $z \in \mathcal{S}'_{i-1}$ , for each  $c \in [q]$  for which  $z \sim f_c$ , we compute

$$x = (z \cup f_c)|_{X_i} \in \text{col}(X_i)$$

and increase  $T'_i[x]$  by  $T'_{i-1}[z]$  if it has been defined already, and initialise it to  $T'_{i-1}[z]$  otherwise. The remaining values are implicitly defined to 0. The running time is as claimed because  $|\mathcal{S}'_{i-1}| \leq (q-1)^{|R_{i-1}|} q^{|X_{i-1}| - |R_{i-1}|}$  and  $||[q]|| \leq q$ .

Next, we compute a set  $R_i$  of reduced vertices for  $T'_i$ . We set  $R_i = X_i \setminus (A_i \cup B_i \cup \{v_i\})$ , where

$$A_i = \{u \in X_i \setminus \{v_i\} : |N(u) \cap Y_i| \geq 2\}$$

and

$$B_i = \{u \in X_i \setminus \{v_i\} : |N(u) \cap Y_i| = 1 \text{ and } uv_i \in E\}.$$

It is easy to see that  $A_i$  and  $B_i$  are disjoint. Within the  $(i-1)$ th cut, each vertex in  $A_i \cup B_i$  has at least two edges going across the cut, so  $|R_i| + 2|A_i| + 2|B_i| \leq \text{ctw}$ . Therefore,  $|X_i \setminus R_i| \leq (\text{ctw} - |R_i|)/2 + 1$ .

We now show that  $R_i$  is indeed a set of reduced vertices. Suppose not, and let  $r \in R_i$  and  $c \in \text{col}(X_i)$  with  $c(r) = q$  yet  $T'_i[c] \neq 0$ . Since  $T'_i[c] \neq 0$ , there exists  $z \in \text{col}(X_{i-1})$  with  $z \sim c$  and  $T'_{i-1}[z] \neq 0$ . By definition  $r \in X_i \setminus \{v_i\} \subseteq X_{i-1}$ . Moreover,  $z(r) = q$  since  $z \sim c$  and  $c(r) = q$ . Therefore  $r$  is not reduced for  $T'_{i-1}$ . We now show  $r$  moreover has degree 1 in the bipartite graph between  $X_{i-1}$  and  $Y_{i-1}$  (corresponding to the  $(i-1)$ th cut), contradicting our assumption that  $T'_{i-1}$  is fully reduced. Since  $r \notin A_i \cup B_i$ , it has at most one edge going over the  $(i-1)$ th cut. Moreover,  $r \in X_i \setminus \{v_i\} \subseteq L_i$ , and so it has at least one edge to  $Y_i \subseteq Y_{i-1}$ . So  $r$  has exactly one neighbor in  $Y_{i-1}$ .

It remains to prove that  $T'_i$  is  $M_i$ -representative of  $T_i$ . This technical part of the proof can be found in the full version of our paper [15].  $\blacktriangleleft$

### 3.4 Analysis of final algorithm

We initialize  $T_1 = \mathbf{1}$ , the all-ones vector. Indeed, each  $x \in \text{col}(\{v_1\})$  has a unique extension to  $G_1$  (namely itself). We then repeatedly apply the **Reduce** algorithm from Lemma 12 until we obtain a fully reduced function  $T'_1$  that is an  $M_1$ -representative of  $T_1$ , with some set of reduced vertices  $R_1$ . For  $i = 2, \dots, n$ , we repeat the following two steps.

1. Apply Lemma 13 with inputs  $(T'_{i-1}, R_{i-1})$  in order to obtain the vector  $T'_i$  that is an  $M_i$ -representative of  $T_i$ , and a set of reduced vertices  $R_i$  for  $T'_i$ .
2. While  $X_i \setminus R_i$  has a vertex  $v$  of degree 1, apply the **Reduce** algorithm from Lemma 12 to  $(T'_i, R_i)$ , and add  $v$  to  $R_i$ .

At the end of step 2, we obtain a fully reduced function  $T'_i$  that is an  $M_i$ -representative of  $T_i$ . Moreover, the set  $R_i$  of reduced vertices has only increased in size compared to the set we obtained in step 1. We apply Lemma 12 at most  $|X_i|$  times in the second step.

## 36:10 Tight Bounds for Counting Colorings and Connected Edge Sets

We eventually compute  $T'_{n-1}$  that is an  $M_{n-1}$ -representative of  $T_{n-1}$  with a fully reduced set  $R_{n-1}$ . We output

$$\sum_{y \in \text{col}(Y_{n-1})} \sum_{x \in \text{col}(X_{n-1})} T'_{n-1}[x] M_{n-1}[x, y].$$

Since  $T'_{n-1}$  is an  $M_{n-1}$ -representative of  $T_{n-1}$ , this gives the number of list colorings of  $G$  modulo  $p$ . We may compute the expression above efficiently by reducing the second summation to the colorings in

$$\mathcal{S}'_{n-1} = \{c \in \text{col}(X_{n-1}) : c(r) \neq q \text{ for all } r \in R_{n-1}\}.$$

The total running time is now bounded by

$$C \sum_{i=1}^{n-1} |X_i| (q-1)^{|R_i|} q^{|X_i| - |R_i|}$$

for some constant  $C > 0$ . By Lemma 13,  $|X_i \setminus R_i| \leq (\text{ctw} - |R_i|)/2 + 1$  for all  $i \in [n-1]$ . For  $q \geq 3$ ,  $q^{1/2} < q-1$  and so

$$(q-1)^{|R_i|} q^{|X_i| - |R_i|} \leq q(q-1)^{|R_i|} (q^{1/2})^{\text{ctw} - |R_i|} < q(q-1)^{\text{ctw}}.$$

This shows the total running time is of order  $(q-1)^{\text{ctw}} n^{O(1)}$ . This finishes the proof of Theorem 8.

### 4 Lower bounds

There exists an efficient reduction from SAT to the problem  $\#_p\text{SAT}$  of counting the number of satisfying assignments for a given boolean formula modulo  $p$  [5]. There also exists a reduction from SAT to  $\text{CSP}(q, r)$ , which preserves the number of solutions [12]. Putting these two together gives a reduction from SAT to  $\#_p\text{CSP}(q, r)$ .

In this section we give a reduction from  $\#_p\text{SAT}$  to  $\#_p\text{LIST } q\text{-COLORING}$ , the problem of counting the number of valid list  $q$ -colorings of a given graph  $G$  with color lists  $(L_v)_{v \in V(G)}$ . We use this to conclude the lower bounds of Theorem 1 and Theorem 2.

#### 4.1 Controlling the number of extensions modulo $p$

Our main gadget can be attached to a given set of vertices, and has the property that for each precoloring of the “glued on” vertices, there is a specified number of extensions. This is made precise in the result below.

► **Theorem 14.** *Let  $k \in \mathbb{N}$  and  $f : [q]^k \rightarrow \mathbb{N}$ . There exist a graph  $G_f$ , a set of vertices  $B = \{b_1, \dots, b_k\} \subseteq V(G_f)$  of size  $k$  and lists  $(L_v)_{v \in V(G_f)}$ , such that for any  $\alpha \in [q]^k$ , there are exactly  $f(\alpha)$  list  $q$ -colorings  $c$  of  $G_f$  with  $c(b_i) = \alpha(i)$  for all  $i \in [k]$ . Additionally,  $|V(G_f)| \leq 20kq^{k+1} \max(f)$  and  $G_f$  has cutwidth at most  $6kq^{k+2}$ .*

The proof is given in Appendix B.

#### 4.2 Reduction for counting $q$ -colorings modulo $p$

In this section we prove the following result.

► **Theorem 15.** *Let  $p$  be a prime and let  $q \in \mathbb{N}$  such that  $p$  does not divide  $q - 1$ . Assuming *SETH*, there is no  $\varepsilon > 0$  for which there exists an algorithm that counts the number of list  $q$ -colorings modulo  $p$  for a given  $n$ -vertex graph, with a given cut decomposition of width  $\text{ctw}$ , in time  $(q - \varepsilon)^{\text{ctw}} n^{O(1)}$ .*

Suppose now that  $p$  divides  $q - 1$ . Let  $q' = q - 1$ . Then  $p$  does not divide  $q' - 1 = q - 2$  and so the result above applies. Noting that any algorithm for  $\#$ LIST  $q$ -COLORING also works for  $\#$ LIST  $q'$ -COLORING, we find the following corollary.

► **Corollary 16.** *Let  $p$  be a prime and let  $q \in \mathbb{N}$  such that  $p$  divides  $q - 1$ . Assuming *SETH*, there is no  $\varepsilon > 0$  for which there exists an algorithm that counts the number of list  $q$ -colorings modulo  $p$  for a given  $n$ -vertex graph, with a given cut decomposition of width  $\text{ctw}$ , in time  $(q - 1 - \varepsilon)^{\text{ctw}} n^{O(1)}$ .*

Combining the two results above with Theorem 8 gives Theorem 1.

We use the notion of *constraint satisfaction problems* (CSP). Informally, a CSP asks if there is an assignment of values from a given domain to a set of variables such that they satisfy a given set of relations. We denote by  $\text{CSP}(q, r)$  the CSP with domain  $[q]$  and constraints of arity at most  $r$ . We use  $\#\text{CSP}(q, r)$  to denote the problem of counting the number of solutions of a given instance of  $\text{CSP}(q, r)$ . We use the following result from [12].

► **Theorem 17** ([12], Theorem 2.5). *For each prime  $p$ , for every integer  $q \geq 2$  and  $\varepsilon > 0$  there is an integer  $r$ , such that the following holds. Unless the *SETH* fails,  $\#_p \text{CSP}(q, r)$  with  $n$  variables and  $m$  constraints cannot be solved in time  $(q - \varepsilon)^n (n + m)^{O(1)}$ .*

This theorem follows from the proof of [12, Theorem 2.5], since their reduction preserves the number of solutions.

**Proof of Theorem 15.** Let  $q \in \mathbb{N}$  and let  $p$  be a prime that does not divide  $q - 1$ . Fix  $\varepsilon > 0$  and let  $r$  be given from Theorem 17. We will reduce a given instance of  $\#_p \text{CSP}(q, r)$  with constraints  $C_1, \dots, C_m$  and variables  $x_1, \dots, x_n$  to an instance  $(G, L)$  of  $\#_p \text{LIST } q$ -COLORING on  $O_{p,r,q}(nm)$  vertices of cutwidth  $n + O_{p,r,q}(1)$ .

The graph  $G$  contains  $2m$  columns with  $n$  vertices: for each constraint  $C_j$ , and for each variable  $x_i$ , we create two vertices  $s_{i,j}$  and  $t_{i,j}$  (where  $j \in [m]$  and  $i \in [n]$ ), which all get  $\{1, \dots, q\}$  as list. For all  $j \in [m - 1]$ , we place an edge between  $t_{i,j}$  and  $s_{i,j+1}$ .

The color assigned to  $s_{i,1}$  will be interpreted as the value assigned to variable  $x_i$ . Fix  $j \in [m]$ . We create gadgets on some vertex set  $V_j$  using Theorem 14, that are “glued” on subsets of vertices from  $C_j = \{s_{i,j}, t_{i,j} : i \in [n]\}$ .

1. For each  $i \in [n]$ , if  $j < m$ , we create a gadget on boundary set  $\{s_{i,j}, t_{i,j}\}$  which ensures that we may restrict to counting list colorings  $c$  of  $(G, L)$  with  $c(s_{i,j}) = c(s_{i,j+1})$ .
2. There is a gadget on a boundary set of size at most  $r$  (the  $s_{i,j}$  corresponding to the variables involved in the  $j$ th constraint), for which the number of extensions of any coloring of the boundary to this gadget is equivalent to 0 modulo  $p$  whenever the  $j$ th constraint is not satisfied, and equal to one otherwise.

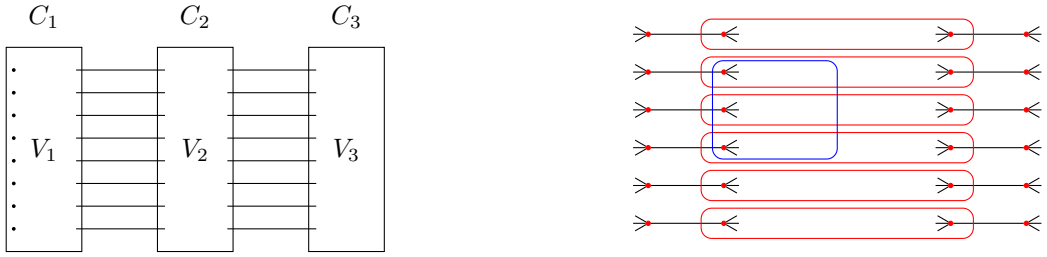
A broad overview of the construction is depicted in Figure 2.

For the first property, we need the fact that  $p$  does not divide  $q - 1$ : this ensures that the color compatibility matrix of a single edge is invertible, which will allow us to “transfer all information about the colors”. The precise construction of the gadgets is deferred to Appendix B.

We obtain a cutwidth decomposition of the graph by first running over the vertices in the order

$$C_1 \cup V_1, C_2 \cup V_2, \dots, C_m \cup V_m.$$

## 36:12 Tight Bounds for Counting Colorings and Connected Edge Sets



■ **Figure 2** A sketch overview of the construction is given on the left-hand side and a more detailed view of two of the columns is given on the right-hand side. The red areas ensure the preservation of information, as described in point 1. The blue area checks whether the clause is satisfied, as described in point 2.

Within  $C_j \cup V_j$ , we first list  $s_{1,j}, t_{1,j}$  and the vertices in the gadget that has those vertices as boundary set, and then repeat this for  $s_{2,j}, t_{2,j}$ , etcetera. Finally, we run over the vertices in the gadget that verifies the  $j$ th constraint. At each point, the cutwidth is bounded by  $n$  plus a constant (that may depend on  $p, q$  and  $r$ ). ◀

### 4.3 Corollaries

We now extend the lower bound of Theorem 15 to counting connected edge sets via the following problem.

► **Definition 18.** Given a graph  $G$ , two  $q$ -colorings  $c$  and  $c'$  are equivalent if there is some permutation  $\pi : [q] \rightarrow [q]$  such that  $c = \pi \circ c'$ . We will refer to these equivalence classes as essentially distinct  $q$ -colorings and denote the problem of counting the number of essentially distinct  $q$ -colorings modulo a prime  $p$  by  $\#_p \text{ESSENTIALLY DISTINCT } q\text{-COLORING}$ .

A simple reduction now gives us the following lower bound for  $\#_p \text{ESSENTIALLY DISTINCT } q\text{-COLORING}$ .

► **Corollary 19.** Let  $p$  be a prime and  $q \in \mathbb{N}$  an integer such that  $p$  does not divide  $q - 1$ . Assuming *SETH*, there is no  $\epsilon > 0$  for which there exists an algorithm that counts the number of essentially distinct  $q$ -colorings mod  $p$  for a given  $n$ -vertex graph that is not  $(q - 1)$ -colorable, with a given cut decomposition of cutwidth  $\text{ctw}$ , in time  $(q - \epsilon)^{\text{ctw}} n^{O(1)}$ .

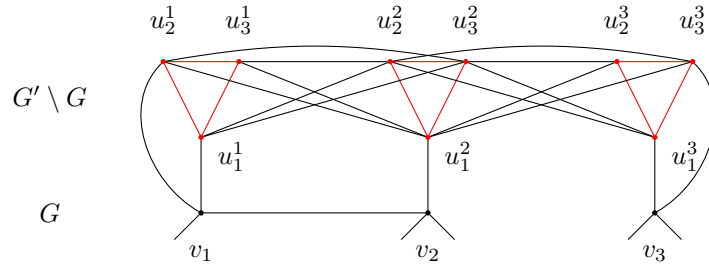
**Proof.** Let  $(G, L)$  be an instance of list coloring with cut decomposition  $v_1, \dots, v_n$ . We construct an instance of  $\#_p \text{ESSENTIALLY DISTINCT } q\text{-COLORING}$ . The graph  $G'$  has vertex set

$$V(G') = V(G) \cup \{u_c^i : c \in [q], i \in [n]\}.$$

We add edges such that the vertices  $\{u_c^i : c \in [q]\}$  induce a  $q$ -clique for all  $i \in [n]$ , and for  $i \in [n - 1]$  we add the edges  $u_c^i u_{c'}^{i+1}$  for all  $c \neq c'$ . This ensures that, if  $u_c^1$  is colored  $c$ , then  $u_c^i$  is colored  $c$  for all  $i \in [n]$ . We now also add edges  $u_c^i u_i$  for all  $c \notin L_{v_i}$ . Our new cut decomposition is

$$v_1, u_1^1, \dots, u_q^1, v_2, u_1^2, \dots, u_q^{n-1}, v_n, u_1^n, \dots, u_q^n.$$

Note that  $\text{ctw}(G') \leq \text{ctw}(G) + q^2$ ,  $|V(G')| \leq (q + 1)|V(G)|$  and that  $G'$  is not  $(q - 1)$ -colorable. By Theorem 1, it suffices to show that the number of essentially distinct  $q$ -colorings of  $G'$  equals the number of list  $q$ -colorings of  $(G, L)$ . We will do this by defining a bijective map.



■ **Figure 3** Example of the construction on (a part of) a graph  $G$ , with the cliques indicated in red. In this case  $q = 3$  and we have  $L_{v_1} = \{3\}$ ,  $L_{v_2} = \{2, 3\}$  and  $L_{v_3} = \{2\}$ .

Let  $\alpha$  be a list coloring of  $(G, L)$ . Then we can color  $G'$  by setting  $\alpha'(v) = \alpha(v)$  for  $v \in V(G)$  and  $\alpha'(u_c^i) = c$  for  $c \in [q]$  and  $i \in [n]$ . This gives us a mapping  $\gamma : \alpha \mapsto \overline{\alpha'}$ , where  $\overline{\alpha'}$  is the equivalence class of  $\alpha'$ . We find an inverse map by first fixing a representative  $\alpha'$  for  $\overline{\alpha'}$ , such that  $\alpha'(u_c^1) = c$  for  $c \in [q]$ . We can do this since  $G'[\{u_1^1, \dots, u_q^1\}]$  is a clique and thus each  $u_c^1$  must get a unique color. Also note that since every color is now used, the rest of the coloring is also fixed and thus we find a unique representative this way. We now map  $\overline{\alpha'}$  to  $\alpha'|_{V(G)}$ . Note that these two maps are well defined and compose to the identity map. We conclude that the number of list colorings of  $(G, L)$  is equal to the number of essentially distinct colorings of  $G'$ . ◀

To achieve the lower bound in Theorem 2, we use an existing argument from [1] to extend this bound to  $\#_p$ CONNECTED EDGE SETS. For this we will need the following definition.

► **Definition 20.** *The  $k$ -stretch of a graph  $G$  is the graph obtained from  $G$  by replacing each edge with a path of length  $k$ . We denote the  $k$ -stretch of  $G$  by  ${}^kG$ .*

Note that  ${}^kG$  has the same cutwidth as  $G$ .

We now show that, assuming SETH, there is no  $\epsilon > 0$  for which there exists an algorithm that counts the number of spanning connected edge sets mod  $p$  of  $n$ -vertex graphs of cutwidth at most  $\text{ctw}$  in time  $O((p - \epsilon)^{\text{ctw}} n^{O(1)})$ .

**Proof.** This proof closely follows a reduction from Annan [1], using ideas from Jaeger, Vertigan and Welsh [17].

Let  $G$  be any graph with cutwidth  $\text{ctw}$  and let  $p$  be a prime. Note that the number of spanning connected edgesets of  $G$  is equal to the value of  $T(G; 1, 2)$ , the Tutte polynomial of  $G$  at the point  $(1, 2)$ . The following formula is found in ([17], proof of Theorem 2)

$$T({}^kG; a, b) = (1 + a + \dots + a^{k-1})^{n-r(G)} T\left(G; a^k, \frac{b + a + \dots + a^{k-1}}{1 + a + \dots + a^{k-1}}\right).$$

Choosing  $a = 1, b = 2$  and  $k = p - 1$ , gives

$$T({}^{p-1}G; 1, 2) = (p - 1)^{n-r(G)} T\left(G; 1, \frac{2 + p - 2}{p - 1}\right) \equiv_p (-1)^{n-r(G)} T(G; 1 - p, 0).$$

Here we use the fact that for any multivariate polynomial  $P(x, y) \equiv_p P(x + tp, y + sp)$  for any  $s, t \in \mathbb{Z}$ . We find that, since the  $k$ -stretch of a graph  $G$  has the same cutwidth as  $G$ , an algorithm that counts the number of spanning connected edgesets (mod  $p$ ) on a graph with bounded cutwidth, also gives the Tutte polynomial at  $(1 - p, 0) \bmod p$ .

## 36:14 Tight Bounds for Counting Colorings and Connected Edge Sets

Using another well known interpretation of the Tutte polynomial [23] we can relate the  $T(G; 1 - p, 0)$  to the chromatic polynomial  $P(G; p)$  as follows:

$$P(G; p) = (-1)^{r(G)} p^{k(G)} T(G; 1 - p, 0).$$

Note that we may assume that the number of connected components  $k(G) = 1$  (and thus  $r(G) = n - 1$ ), since the number of spanning connected edgesets is trivially 0 if  $G$  has more than one component. We want to get rid of the remaining factor of  $p$  (since we will work mod  $p$ ). To do this we will count the number of essentially distinct colorings (different up to color-permutations) using exactly  $p$  colors instead. This will turn out to be at least as hard as counting all colorings.

Let  $G$  be a graph that is not  $(p - 1)$ -colorable. With this assumption, the number of  $p$  colorings of  $G$  is  $p!$  times the number  $C_p(G)$  of essentially distinct  $p$ -colorings of  $G$ , since any coloring uses all colors and thus can be mapped to  $p!$  equivalent colorings by permuting the colors. So

$$(-1)^{n-1} p T(G, 1 - p, 0) = P(G; p) = p(p - 1)! C_p(G),$$

which holds over the real numbers hence we may divide both sides by  $p$ . By Wilson's Theorem  $(p - 1)! \equiv_p -1$ , so we find

$$(-1)^{n-1} T(G, 1 - p, 0) \equiv_p -C_p(G).$$

Hence we can use the number of spanning connected edgesets (mod  $p$ ) of the  $(p - 1)$ -stretch of  $G$  to find the Tutte polynomial at  $(1 - p, 0) \bmod p$  and then also the number of essentially distinct colorings. The lower bound of Theorem 2 now follows from Corollary 19 (with  $q = p$ ). The upper bound is proved in the full version of our paper [15]. ◀

## 5 Conclusion

In this paper we give tight lower and upper bounds for counting the number of (list)  $q$ -colorings and connected spanning edge sets of graphs with a given cutwidth decomposition of small cutwidth. Our results specifically relate to list  $q$ -coloring and essentially distinct  $q$ -coloring, but they can easily be extended to normal  $q$ -coloring for certain cases. In particular, if  $q < p$ , we may apply Corollary 19, since in the setting of the corollary, the values differ by  $q!$  which is nonzero modulo  $p$ . If the chromatic number  $\chi(G) \geq p$ , then the number of  $q$ -colorings is trivially 0 mod  $p$ , since the number of  $q$ -colorings is a multiple of  $\chi(G)$ . This leaves us with the rather specific case of  $\chi(G) < p \leq q$ , for which the exact complexity remains unresolved.

Our results on the modular counting of colorings show that the modulus can influence the complexity in interesting ways, and that in some cases this effect can be directly explained by the rank of the compatibility matrix.

Our results leave several directions for further research:

- What is the fine-grained complexity of evaluating other points of the Tutte polynomial (modulo  $p$ )?
- What is the complexity of counting homomorphisms to graphs different from complete graphs, e.g. cycles or paths. Is it still determined by the rank of an associated compatibility matrix?

Another question in the direction of fast *decision problems* is how small representative sets we can get for the compatibility matrix of graphs other than complete bipartite graphs (which is equivalent to the setting of [13, Theorem 1.2]) or matchings which has been studied for the decision version in [18].

## References

- 1 James Douglas Annan. *The complexity of counting problems*. PhD thesis, University of Oxford, 1994.
- 2 Andreas Björklund and Thore Husfeldt. Inclusion-exclusion based algorithms for graph colouring. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 13. Citeseer, 2006.
- 3 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Computing the tutte polynomial in vertex-exponential time. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 677–686. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.40.
- 4 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.
- 5 Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-sat: An isolation lemma for k-cnfs. *Journal of Computer and System Sciences*, 74(3):386–393, 2008.
- 6 Radu Curticapean, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting hamiltonian cycles via matrix rank. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1080–1099. SIAM, 2018. doi:10.1137/1.9781611975031.70.
- 7 Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016. doi:10.1137/1.9781611974331.ch113.
- 8 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.
- 9 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Joham MM van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 150–159. IEEE, 2011.
- 10 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the tutte polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014. doi:10.1145/2635812.
- 11 Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures & Algorithms*, 17(3-4):260–289, 2000.
- 12 Jacob Focke, Dániel Marx, and Paweł Rzażewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 431–458. SIAM, 2022.
- 13 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.
- 14 Catherine Greenhill. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. *Computational Complexity*, 9(1):52–72, 2000.
- 15 Carla Groenland, Jesper Nederlof, Isja Mannens, and Krisztina Szilágyi. Tight bounds for counting colorings and connected edge sets parameterized by cutwidth. *arXiv preprint*, 2021. arXiv:2110.02730.
- 16 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 17 François Jaeger, Dirk L Vertigan, and Dominic JA Welsh. On the computational complexity of the Jones and Tutte polynomials. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 108(1), pages 35–53. Cambridge University Press, 1990.

- 18 Bart MP Jansen and Jesper Nederlof. Computing the chromatic number using graph decompositions via matrix rank. *Theoretical Computer Science*, 795:520–539, 2019.
- 19 Amirhossein Kazemini and Andrei A Bulatov. Counting homomorphisms modulo a prime number. *arXiv preprint*, 2019. [arXiv:1905.10682](https://arxiv.org/abs/1905.10682).
- 20 Mikko Koivisto. An  $o^*(2^n)$  algorithm for graph coloring and other partitioning problems via inclusion–exclusion. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 583–590. IEEE Computer Society, 2006. [doi:10.1109/FOCS.2006.11](https://doi.org/10.1109/FOCS.2006.11).
- 21 Daniel Lokshтанov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. [doi:10.1145/3170442](https://doi.org/10.1145/3170442).
- 22 Jesper Nederlof. Bipartite TSP in  $O(1.9999^n)$  time, assuming quadratic time matrix multiplication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 40–53. ACM, 2020. [doi:10.1145/3357713.3384264](https://doi.org/10.1145/3357713.3384264).
- 23 James G Oxley, Dominic JA Welsh, et al. The tutte polynomial and percolation. *Graph Theory and Related Topics*, pages 329–339, 1979.
- 24 Leslie G. Valiant. Accidental algorithms. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 509–517. IEEE Computer Society, 2006. [doi:10.1109/FOCS.2006.7](https://doi.org/10.1109/FOCS.2006.7).
- 25 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. [doi:10.1007/978-3-642-04128-0\\_51](https://doi.org/10.1007/978-3-642-04128-0_51).

## A Proof of Lemma 12

Let  $H$  a bipartite graph with parts  $X, Y$  and associated color compatibility matrix  $M$ . Let  $T : \text{col}(X) \rightarrow \mathbb{F}_p$  be a function with reduced vertices  $R \subseteq X$  and let vertex  $v \in X \setminus R$  be a vertex of degree 1 in  $H$ . We need to find a function  $T' : \text{col}(X) \rightarrow \mathbb{F}_p$  with reduced vertices  $R \cup \{v\}$  that is  $M$ -representative of  $T$ . (And need to show this can be done in time  $O((q-1)^{|R|}q^{|X|-|R|})$ .)

We may restrict to colorings  $x$  that do not assign value  $q$  to any element of  $R$ . There are at most  $(q-1)^{|R|}q^{|X \setminus R|}$  such colorings. We set

$$T'[x] = \begin{cases} 0, & \text{if } x(v) = q, \\ T[x] - T[x'], \text{ where } x' \text{ is obtained from } x \text{ by changing the value of } v \text{ to } q, & \text{otherwise.} \end{cases}$$

This computation is done in time linear in the number of the colorings  $x$  we consider, so the running time is in  $O((q-1)^{|R|}q^{|X \setminus R|})$ .

First we will show that  $T'$  is an  $M$ -representative of  $T$ . Let  $y \in \text{col}(Y)$  be a coloring of the right hand side of the bipartite graph  $H$ . We need to show that

$$\sum_{\substack{x \in \text{col}(X) \\ x \sim y}} T[x] \equiv_p \sum_{\substack{x \in \text{col}(X) \\ x \sim y}} T'[x].$$

By definition,

$$\sum_{\substack{x \in \text{col}(X) \\ x \sim y}} T'[x] = \sum_{\substack{x \in \text{col}(X) \\ x \sim y \\ x(v)=q}} 0 + \sum_{\substack{x \in \text{col}(X) \\ x \sim y \\ x(v) \neq q}} T[x] - T[x'].$$



Thus it remains to show that

$$\sum_{\substack{x \in \text{col}(X) \\ x \sim y \\ x(v) \neq q}} -T[x'] \equiv_p \sum_{\substack{x \in \text{col}(X) \\ x \sim y \\ x(v) = q}} T[x].$$

Let  $x \in \text{col}(X)$  with  $x(v) = q$ . We show the equality by proving that the number of times  $T[x]$  appears on the left hand side equals the number of times  $T[x]$  appears on the right hand side, modulo  $p$ . Let  $w \in Y$  be the unique neighbor of the vertex  $v$ .

First assume that  $x \sim y$ . Then  $y(w) \neq q$ . If we adjust  $x$  to the coloring  $x_i$ , which is equal to  $x$  apart from assigning color  $i$  to  $v$  instead of  $q$ , then  $x_i \sim y$  if and only if  $i \neq y(w)$ . Hence the term  $-T[x]$  appears  $q - 2$  times on the left hand side, and  $T[x]$  appears once on the right hand side. Since  $p$  divides  $q - 1$ , we find  $q - 2 \equiv_p -1$  and hence both contributions are equal modulo  $p$ .

If  $x \not\sim y$ , then either  $x$  does not appear on both sides (because  $x|_{X \setminus \{v\}}$  is already incompatible with  $y$ ) or  $y(w) = q$ . If  $y(w) = q$ , then the term  $T[x]$  appears  $q - 1 \equiv_p 0$  times on the left hand side by a similar argument as the above, and does not appear on the right hand side. This shows the claimed equality and finishes the proof.

## B Proofs omitted from Section 4

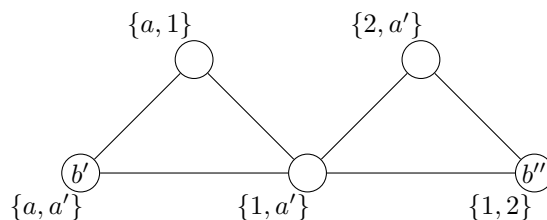
### B.1 Proof of Theorem 14

We first reduce the lists of each  $b_i$  to  $\{1, 2\}$  using the following gadget.

► **Lemma 21.** *Let  $q, k \in \mathbb{N}$  and let  $a \in [q]$ . There is a graph  $G$  with  $b, b' \in V(G)$  and color lists  $L_v \subseteq [q]$  for  $v \in V(G)$ , such that  $L_b = [q]$  and the following two properties hold:*

- for all  $c_b \in [q]$ , there is a unique list coloring  $c$  of  $G$  with  $c(b) = c_b$ ,
- for all list colorings  $c$  of  $G$ , if  $c(b) = a$ , then  $c(b') = 1$  and if  $c(b) \neq a$  then  $c(b') = 2$ .

**Proof.** We first note that it is easy to “relabel colors”, as shown in the construction<sup>3</sup> in Figure 4. We can therefore first make a gadget for which  $b'$  has color list  $\{a, a'\}$  for some



■ **Figure 4** A gadget to “relabel colors”. It has two special vertices  $b'$  and  $b''$ , and lists are depicted with sets. For any list coloring  $c$  of the depicted gadget, if  $c(b') = a$ , then  $c(b'') = 1$  and if  $c(b') = a'$ , then  $c(b'') = 2$ . In both cases, there is a unique way to color the remaining vertices.

$a' \neq a$ , and then relabel  $a, a'$  to  $1, 2$ . By symmetry, we can therefore assume that  $a = 1$  (or simply replace  $1$  with  $a$  and  $2$  with  $a'$  in the argument below). Let

$$V = \{b, b'\} \cup \{s_i : i = 2, \dots, q\} \cup \{t_i : i = 2, \dots, q\}$$

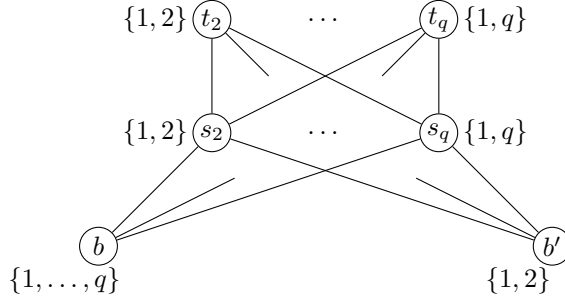
and

<sup>3</sup> If  $a = 1$  or  $a' = 2$  we slightly change the construction by removing the top left or top right vertex respectively.

### 36:18 Tight Bounds for Counting Colorings and Connected Edge Sets

$$E = \{s_i b : i = 2, \dots, q\} \cup \{s_i b' : i = 2, \dots, q\} \cup \{s_i t_j : i, j = 2, \dots, q\}.$$

Now let  $L_b = [q]$ ,  $L_{b'} = \{1, 2\}$  and  $L_{t_i} = L_{s_i} = \{1, i\}$  for  $i \in \{2, \dots, q\}$ . A depiction is given in Figure 5.



■ **Figure 5** The construction of the list coloring instance of the proof of Lemma 21.

If a list coloring  $c$  of  $G$  satisfies  $c(b) = 1$ , then  $c(s_i) = i$  and thus  $c(t_i) = 1$  for each  $i \in \{2, \dots, q\}$ . In particular  $c(s_2) = 2$  and  $c(b') = 1$ .

If  $c_b \in \{2, \dots, q\}$ , then any list coloring  $c$  with  $c(b) = c_b$  satisfies  $c(s_i) = 1$  and  $c(t_i) = i$  for all  $i \in \{2, \dots, q\}$ , and so  $c(b') = 2$ .

This proves that, starting with the color  $c_b \in [q]$  for  $b$ , there is always a unique extension to a list coloring of  $G$ , and this satisfies the property that vertex  $b'$  receives color 1 if  $c_b = 1$ , and receives color 2 otherwise. ◀

We also make use of the following construction.

► **Lemma 22.** *Let  $k, \ell \in \mathbb{N}$ . There is a graph  $G$ , a subset of vertices  $B = \{b_1, \dots, b_k\} \subseteq V$  of size  $k$ , and color lists  $L_v$  for all  $v \in V(G)$  such that:*

- $L_{b_i} = \{1, 2\}$  for all  $i \in \{1, \dots, k\}$ ,
- there are exactly  $\ell$  list colorings  $c$  of  $G$  with  $c(B) = \{1\}$ ,
- for each partial coloring  $c_B$  of  $B$  with  $c_B(B) \neq \{1\}$ , there is a unique extension of  $c_B$  to a list coloring of  $G$ .

**Proof.** We start with  $V = B$  and add a path<sup>4</sup>  $w_1, \dots, w_{\ell-1}$  with color lists

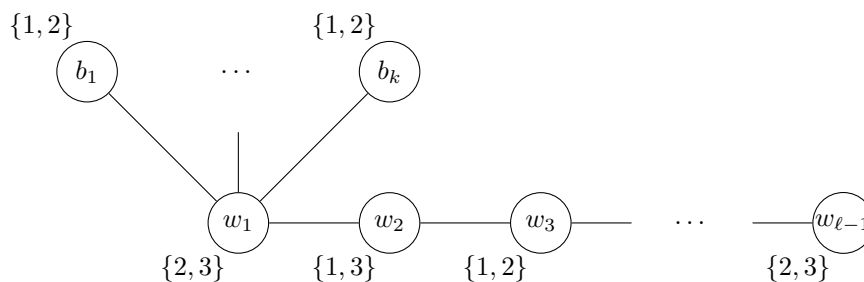
$$L_{w_i} = \begin{cases} \{2, 3\} & \text{if } i \equiv_3 1, \\ \{1, 3\} & \text{if } i \equiv_3 2, \\ \{1, 2\} & \text{if } i \equiv_3 0, \end{cases}$$

and add edges  $b_i w_1$  for  $i = 1, \dots, k$ . A depiction is given in Figure 6.

If a list coloring  $c$  satisfies  $c(b_i) = 2$  for some  $i \in [k]$ , then  $c(w_1) = 3$ ,  $c(w_2) = 1$ ,  $c(w_3) = 2$  etcetera. Hence there is a unique extension of any partial coloring of  $B$  that assigns color 2 somewhere.

If  $c(b_i) = 1$  for all  $i \in [k]$ , then we have a choice for the color of  $w_1$ . If  $c(w_1) = 3$  then we get the same propagation as before, however if  $c(w_1) = 2$ , then we have a choice for the color of  $w_2$ . Using a simple induction argument we find that the number of possible list colorings with  $c(B) = \{1\}$  equals  $\ell$ . ◀

<sup>4</sup> When  $\ell = 1$ , we add no vertices of the form  $w_i$  and the statements of the lemma immediately follow.



■ **Figure 6** Construction in the proof of Lemma 22 when  $\ell \equiv_3 2$ .

We are now ready to construct the main gadget.

**Proof of Theorem 14.** Let  $f \in [q]^k$ . We create vertices  $b_1, \dots, b_k$  and give them all  $[q]$  as list.

For each partial coloring  $\alpha \in [q]^k$ , we create a graph  $G_\alpha$  that contains  $b_1, \dots, b_k$  in their vertex set, but the graphs are on disjoint vertex sets otherwise (we “glue” the graphs on the special vertices  $b_1, \dots, b_k$ ). It suffices to show that we can find lists for the “private” vertices of  $G_\alpha$  such that the number of extensions of a coloring  $c_B$  of  $B$  is 1 if  $c_B(b_i) \neq \alpha(i)$  for some  $i \in [k]$ , and  $f(\alpha)$  otherwise. The resulting gadget will then have  $1 \cdot 1 \cdot \dots \cdot 1 \cdot f(\alpha) = f(\alpha)$  possible extensions for the precoloring  $\alpha$ , as desired.

We now turn to constructing the gadget  $G_\alpha$  for a fixed coloring  $\alpha \in [q]^k$ . We first reduce to the case in which each  $b_i$  has  $\{1, 2\}$  as list. Let  $i \in [k]$ . Using Lemma 21 with  $a = \alpha(i)$ , we obtain a gadget  $H_{b,b'}$  and identify the special vertex  $b$  with  $b_i$ . For each  $\alpha$ , we obtain a new set of vertices  $b'_1, \dots, b'_k$  with lists  $\{1, 2\}$ . We then glue these onto the special vertices from a gadget obtained by applying Lemma 22 with  $\ell = f(\alpha)$ . If  $b_1, \dots, b_k$  are colored as specified by  $\alpha$ , then  $b'_1, \dots, b'_k$  all receive color 1 and  $G_\alpha$  has  $f(\alpha)$  possible extensions; however if some  $b_i$  receives the wrong color, the corresponding  $b'_i$  receives color 2 and there is a unique extension to the rest of  $G_\alpha$ .

It remains to show the bounds on the number of vertices and the cutwidth. We give the very rough upperbound of  $6kq^{k+2}$  on the cutwidth. The gadget from Lemma 21 has cutwidth at most  $q^2 + 6$  (since this is an upper bound on the number of edges in that construction). The gadgets from Lemma 22 have cutwidth at most  $k$ . A final cut decomposition can be obtained by first enumerating the vertices in  $B$ , and then adding the cut decompositions of each  $G_\alpha$ , one after the other.

Finally the number of vertices of the graph is upper bounded by  $q^k$  times the maximum number of vertices of the graph  $G_\alpha$ . The gadget of Lemma 21 has at most  $2q + 6$  vertices and there are  $k$  of them, so they contribute at most  $12kq$  vertices. The gadgets from Lemma 22 add at most  $f(\alpha)$  vertices. In total,

$$|V(G_f)| \leq 20kq^{k+1} \max(f). \quad \blacktriangleleft$$

## B.2 Remaining details of the proof of Theorem 15

We first describe the gadgets for the “color transfer” (the first desired property). Let  $j \in [m - 1]$  and  $i \in [n]$ . We will apply Theorem 14 to a function  $f_{i,j}$  with boundary set  $B_{i,j} = (s_{i,j}, t_{i,j})$  and  $\max(f_{i,j}) = p$ , resulting in a graph on  $O_{p,q}(1)$  vertices. Let  $J_1$  be the  $q \times q$  coloring compatibility matrix of a single edge, and let  $J_1^{-1}$  denote its inverse over  $\mathbb{F}_p$

## 36:20 Tight Bounds for Counting Colorings and Connected Edge Sets

(that is,  $J_1^{-1}J_1 \equiv_p I_q$ , the  $q \times q$  identity matrix). We “choose a representative”  $\widetilde{J}_1$ , which has entries in  $\{1, \dots, p\}$  that are equivalent to those in  $J_1^{-1}$  modulo  $p$ . For  $c_1, c_2 \in [q]$  possible colors for  $s_{i,j}$  and  $t_{i,j}$  respectively, we set

$$f_{i,j}(c_1, c_2) = \widetilde{J}_1[c_1, c_2].$$

Let  $V_{i,j}$  denote the vertices in the gadget obtained by applying Theorem 14 to  $(f_{i,j}, B_{i,j})$  that are not in  $B_{i,j}$ . Let  $c_1, c_3 \in [q]$ . The number of list colorings  $c$  of the graph induced on  $B_{i,j} \cup V_{i,j} \cup \{s_{i,j+1}\}$  with  $c(s_{i,j}) = c_1$  and  $c(s_{i,j+1}) = c_3$  is equal to

$$\sum_{c_2 \in [q]} f_{i,j}(c_1, c_2) J_1[c_2, c_3] = (\widetilde{J}_1 J_1)[c_1, c_3],$$

since for any coloring  $c_2$  we have  $f_{i,j}(c_1, c_2) J_1[c_2, c_3]$  such colorings with  $c(t_{i,j}) = c_2$ , by definition of  $f_{i,j}$  and  $J_1$ . Therefore, modulo  $p$  this number of extensions is equal to 1 if  $c_1 = c_3$  and 0 otherwise, as desired.

We now describe the gadgets that check the constraints. Let  $j \in [m]$  and let  $i_1, \dots, i_\ell$  be given so that the  $j$ th constraint only depends on the variables  $x_{i_1}, \dots, x_{i_\ell}$  (where by assumption  $\ell \leq r$ ). We will apply Theorem 14 to a function  $g_j$  with boundary set  $B_j = (s_{i_1,j}, \dots, s_{i_\ell,j})$  and  $\max(g_j) = p$ , resulting in a graph on  $O_{p,q,\ell}(1) = O_{p,q,r}(1)$  vertices. We set  $g_j(c_1, \dots, c_\ell)$  to be equal to 1 if the assignment  $(c_1, \dots, c_\ell)$  to  $(x_{i_1}, \dots, x_{i_\ell})$  satisfies the  $j$ th constraint, and  $p$  otherwise. This ensures the second property described in the proof of Theorem 15.